Uniwersytet WSB Merito w Poznaniu

Wydział Finansów i Bankowości

Studia niestacjonarne I stopnia – Informatyka

Przedmiot: **Programowanie zaawansowane**

Grupa: **K32 W09**

Rok akad.: **2024/2025**

Prowadzący: dr Paweł Płaczek

Planowanie grafiku ekipy sprzątającej

Opis projektu

Program służy do zarządzania grafikami osób sprzątających. Projekt umożliwi rejestrację dwóch typów użytkowników: administratora i użytkownika. Administrator ma pełny dostęp do wszystkich lokalizacji i pracowników. Użytkownik otrzymuje dostęp wyłącznie do przypisanych mu lokalizacji, którymi może zarządzać. Każda lokalizacja ma przypisanych pracowników, którzy są przydzielani do grafiku sprzątania.

Autorzy:

-Adam Orzechowski

-Krzysztof Oziembłowski

Specyfikacja Wykorzystanych technologii

HTTPS- Protokół komunikacyjny używany w sieci do bezpiecznego przesyłania danych. Wykorzystuje TLS do szyfrowania informacji między klientem a serwerem, zapewniając poufność, integralność i autentyczność danych. Używany przede wszystkim w aplikacjach internetowych do ochrony prywatnych informacji użytkowników.

SQLite- Lekka, wbudowana baza danych SQL, która przechowuje dane w pojedynczym pliku na dysku. Nie wymaga osobnego serwera co czyni ją idealnym rozwiązaniem dla aplikacji o niskiej skali lub lokalnych zastosowań. SQLite obsługuje

standard SQL, ale ma ograniczenia w porównaniu do systemów zarządzania bazami danych takich jak MySQL czy PostgreSQL.

.NET 8.0- najnowsza wersja platformy .NET firmy Microsoft, która umożliwia budowanie aplikacji wieloplatformowych (webowych, mobilnych, desktopowych, IoT) Oferuje wydajność, bezpieczeństwo i narzędzia do nowoczesnych technologii, takich jak sztuczna inteligencja i mikroserwisy. Wersja 8.0 wprowadza usprawnienia w obsłudze HTTP/3, lepszą integrację z chmurą oraz wsparcie dla innowacyjnych funkcji języka C#.

ASP.NET Core MVC- Framework wchodzący w skład platformy .NET, używany do tworzenia aplikacji internetowych w modelu MVC. Pozwala na tworzenie dynamicznych stron internetowych i API, oddzielając logikę aplikacji (Model) od jej prezentacji (View) i kontroli (Controller). ASP.NET Core jest wieloplatformowy, lekki i zoptymalizowany pod kątem wydajności.

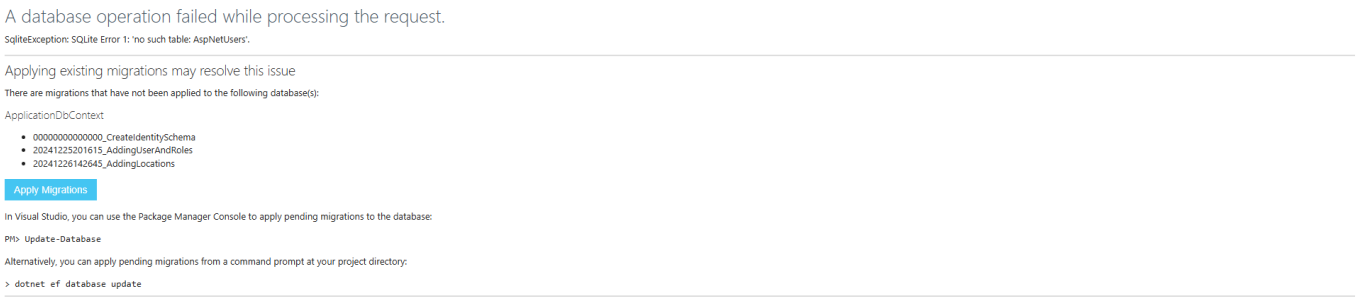
Entity Framework- Biblioteka ORM dla platformy .NET, umożliwiająca programistom pracę z bazami danych za pomocą kodu obiektowego, zamiast bezpośredniego pisania zapytań SQL. Ułatwia tworzenie, modyfikowanie i zarządzanie danymi w aplikacjach .NET, wspierając różne bazy danych, wtym SQLite, SQL Server, i PostgreSQL.

Instrukcję pierwszego uruchomienia

1.Otworzenie projektu MVC

2.Uruchomienie aplikacji przyciskiem Run lub F5

3. Stworzenie migracji przyciskiem “Apply Migrations”



4. Zalogowanie się za pomocą danych

Administrator

Login: admin@localhost Hasło: admin

Użytkownik

Login: user@localhost Hasło: user

Opis struktury projektu

Zastosowanie wzorca MVC:

- Struktura projektu oparta na wzorcu Model-View-Controller, co zapewnia przejrzystość oraz separacje warstw odpowiedzialnych za danę.

Definiowane ról użytkowników w CommonRoles:

W celu eliminacji “magic string” w kodzie, role użytkowników zostały zdefiniowane jako stałe dedykowanej klasie CommonRoles

Fabryka ApplicationUserPrincipalFactory w katalogu services

Wylistowane wszystkie modele

ApplicationUser.cs

Location.cs

ErrorViewModel.cs

Wylistowane kontrolery wraz z metodami

HomeController.cs

- Index() - wyświetla stronę główną aplikacji
- Privacy() - wyświetla stronę polityki prywatności
- Error() - wyświetla stronę błędu

LocationsController.cs

- Index ()
- MyLocations()
- Details(Guid? id)
- Create()
- Create(Location location, IFromFile? Image)
- Edit(Guid? Id)
- Edit(Guid id, Location location)
- Delete(Guid? Id)

- DeleteConfirmed(Guid id)
- LocationsExists(Guid id)

UserMenagmentController.cs

- Index()
- Edit(Guid? Id)
- Edit(Guid id, UserWithRolesViewModel roles)

EmployeesController.cs

- Index()
- Details(Guid? id)
- Create()
- Create(Employee employee)
- Edit(Guid? id)
- Edit(Guid id, Employee employee)
- Delete(Guid? id)
- DeleteConfirmed(Guid id)
- EmployeeExists(Guid id)

Opis systemu użytkowników

System przewiduje dwie role Administrator i użytkownik.

Administrator:

Posiada pełne uprawnienia w systemie

Zarządza użytkownikami, w tym ma możliwość przypisania do nich roli

Możliwość zarządzania lokalizacjami, dodawanie/edytowanie lokalizacji, oraz przypisywanie do nich pracowników

Użytkownik
posiada dostęp do harmonogramu, jedynie tej lokalizacji do której jest dopisany Jeśli nie jest dopisany do żadnej lokalizacji lub nie ma przypisanej roli nie ma dostępu do funkcjonalności

Najciekawsze funkcjonalności

- Rozszerzona klasa użytkowników o dodatkowe pola (Imię nazwisko obrazek),

- Dodanie do projektu user claims zawierające dodatkowe pola użytkownika aby można było wygodnie korzystać z nich na widokach,
- Zastosowanie gotowego UI frameworku Hyper do poprawy estetyki ,
- W common roles zdefiniowano role użytkowników, aby w kodzie aplikacji nie używać “magic strings”.